(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 325 443 A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89300480.4

(22) Date of filing: 19.01.89

(51) Int. Cl.4: G 06 F 3/023

(30) Priority: 22.01.88 US 146985

(43) Date of publication of application:
26.07.89 Bulletin 89/30

(84) Designated Contracting States: DE FR GB

(71) Applicant: Hewlett-Packard Company
3000 Hanover Street
Palo Alto California 94304 (US)

(72) Inventor: Weiner, Jonathan A.
726 Channing Avenue
Palo Alto California 94301 (US)

(74) Representative: Colgan, Stephen James et al
CARPMAELS & RANSFORD 43 Bloomsbury Square
London WC1A 2RA (GB)

(54) Help facility shared by a plurality of applications.

(57) A help system is presented which provides a uniform interface for a plurality of applications (11,12,13). For each application in the plurality of applications (11,12,13) a help text file (82) is developed. From each application in the plurality of applications, a user may call a main help routine (16). This main help routine (16) accesses the help text file (82) for the application from which the main help routine is called. Using this help text file (82) the main help routine (16) displays text which gives a user information about the application the user is currently using. Thus a user is able to use a standard interface to obtain help information about a particular application.

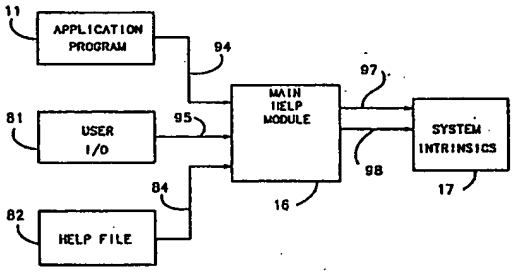

FIG 10

EP 0 325 443 A2

Description

HELP FACILITY SHARED BY A PLURALITY OF APPLICATIONS

Background

On-line help systems are facilities that are an integral part of many interactive applications. On-line help systems provides to a user information about how to use an application while the user is using the application. Such on-line help systems are especially helpful to a user who does not want to take time to look up information in a manual, a user who does not have a manual, or a user who is intimidated by manuals.

Typically, however, help systems are added to an application hastily, as the last thing to be done before an application goes "out the door". The result is often a help system that is incomplete, poorly written or confusing to a user.

Summary of the Invention

In accordance with the preferred embodiment of the present invention, a help system is presented which provides a uniform interface for a plurality of applications. For each application in the plurality of applications a help text file is developed. From each application in the plurality of applications, a user may call a main help routine. This main help routine accesses the help text file for the application from which the main help routine is called. Using this help text file the main help routine displays text from the help text file. Thus a user is able to use a standard interface to obtain help information about a particular application.

Brief Description of the Drawings

Figure 1 shows a block diagram of a computing system in accordance with the preferred embodiment of the present invention.

Figure 2 shows a mouse in accordance with the preferred embodiment of the present invention.

Figures 3 through 9 show display screens which illustrate performance of the preferred embodiment of the present invention.

Figures 10 through 16 are block diagrams which show information flow through the preferred embodiment of the present invention.

Figures 17 and 18 show the structure of a file used in the preferred embodiment of the present invention.

Description of the Preferred Embodiment

Figure 1 shows a computing system in accordance with the preferred embodiment of the present invention. A monitor 14, a mouse 20 and a keyboard 19 are coupled to a computer 18. A memory 10 is resident within computer 18 as represented by arrow 21.

Memory 10, may, for example, contain an application program 11, an application program 12, an application program 13, a main help module 16 and system intrinsics 17.

Application programs 12, 13 and 14 call a main help module 16. Main help module 16 utilizes system intrinsics 17 to display information on monitor 14.

Figure 2 shows a top view of mouse 20. Mouse 20 includes a button 27 and a button 28.

Figure 3 shows a window 1 displayed upon monitor 14. Window 1 includes a command bar 3. Command bar 3 includes a list of commands. For example, "Edit" "Transfer" and "Help" are shown. The command "Help" is listed in command bars displayed by applications 11, 12 and 13, and any other application that calls main help module 16.

A cursor 2, controlled by mouse 20, is used to select commands on command bar 3. For instance, placing cursor 2 over "Help" on command bar and clicking button 27 on mouse 20 results in the appearance of a menu 4 shown in Figure 4.

As cursor 2 points at an entry in menu 4 causes the entry to be shaded. Clicking button 27 on mouse 20 results in the shaded entry being selected. For instance, if "Cancel" is chosen, window 20 will return to the display shown in Figure 3. In Figure 4, the entry "Index" is shown shaded. Clicking button 27 on mouse 20 would result in the display of a selection box 20 as shown in Figure 5.

Within selection box 30 is a list box 34. Within list box 34 selection entries 33 are displayed. In the present embodiment, each entry from selection entries 33 describes a topic which for which a "help" program will provide information. Thus, for instance, a user can get information on icons by selecting the entry from the selection of displayed entries 33 which says "Icon".

When a user wants to select one of selection 33 he moves mouse 20 such that cursor 2 is placed over a selected entry of selection entries 33. The user then pushes down button 27 on mouse 20, which causes the selected entry to be highlighted. Moving mouse 20 and thus cursor 2 will cause different entries to be highlighted. Once a user releases button 27 on mouse 20, the highlighted entry will be selected. When a entry is selected, information pertaining to the selected entry will be displayed.

Only a limited number of selections may be displayed at one time. For instance, in Figure 5, only nine entries in alphabetical order are shown. However, a total list of the entries may be much larger. The other selections may be addressed by using scroll bar 32. Scroll bar 32 contains a scroll region 37, an arrow box 35 and an arrow box 36. In Figure 5 scroll region 37 is at the border of arrow box 35. This indicates that the first entries of the list of entries are shown. When scroll region 37 is at the border of

arrow box 36, this indicates that the last entries of the list of entries are shown. When scroll region 37 is in between these extreme positions, this indicates some intermediate entries of the list of entries are being shown.

Scroll region 37 may be moved by various methods. For instance, when scroll region 37 is not at the border of arrow box 36, a user may scroll down the shown list of entries by placing cursor 2 in arrow 36 and depressing button 27 on mouse 20. Upon each depression of the button the top displayed entry disappears, a new bottom entry appears at the bottom and all the other selections move upward. Also, scroll region 37 moves incrementally downward. Similarly, placing cursor 2 in arrow 35 and depressing the button on the mouse causes the list of entries to be scrolled upward and scroll region 37 to be moved incrementally upward. Further, the list of entries may be scrolled by placing cursor 2 in scroll regions 37 and depressing the button on the mouse. The scroll region will then follow cursor 2 up and down scroll bar 32 until button 27 is released. Upon release, entries in the list of entries would appear which would correspond to the location of scroll region 37 on scroll bar 32.

Additionally shown in Figure 5 is an edit box 31. Edit box 31 may be used to type in the name of the selection. This allows the user to make a selection using keyboard 19, and without using mouse 30. For instance, rather than using a mouse to select the entry labelled "Close", a user could, using keyboard 19, type in the letters "c" "l" "o" "s" "e". The word "close" appear in edit box 31. Upon pressing "ENTER" on keyboard 19, a search control processor within main help module 16 would cause the entry "Close" to be selected.

Letters typed into edit box 31 also interacts with displayed entries 33 in list box 34 to allow for a shorthand method to select entries. When a first letter is typed into edit box 31, the search control processor causes the first selection in list box with the same first letter to be highlighted. For instance, if the letter "s" is typed in edit box 31, the entry "Select Item" would be highlighted. If a user presses "ENTER" on keyboard 19, the search control processor will cause the entry "Select Item" to be selected.

If instead the user types another letter, the search control processor causes the first entry in list box 34 with the same first two letters to be highlighted. For instance, if the user typed in the letter "e", the entry "Select Item" would remain highlighted. If, instead, the user typed in the letter "h", the entry "Shadow" would be highlighted. If the user now presses "ENTER" on keyboard 19, the search control processor will cause the entry "Shadow" to be selected.

This process continues for every letter typed into edit box 31. If at this point the user presses "BACKSPACE" on keyboard 19, the "h" will disappear from edit box 31. Additionally, the highlighting in list box 34 will match the entry in edit box 31, so that the entry "Shadow" will no longer be highlighted, but instead the entry "Select Item" will be highlighted. If the user presses "ENTER" on keyboard 19, the search control processor will cause the entry "Select Item" to be selected.

If the list of entries is larger than may be displayed within list box 34, this does not prevent non-displayed entries from being selected. The search control processor will continue to search through non-displayed entries looking for a match. If a match is found the search control processor will scroll displayed entries 33 to display the entry that matches.

If there is no entry that matches, the search control processor will scroll list box 34 to the top of the list of entries and no entries will be highlighted. If the user presses "ENTER" nothing will happen. An alternate embodiment could allow for an error message to be displayed.

If instead of "Index", a user highlighted entry "Select Item" of menu 5, as shown in Figure 6, upon releasing button 27, a special help cursor 6 would appear as shown in Figure 7. A user may use Help cursor 6 to find out information about any command in command bar 3. For instance, if a user uses help cursor 6 to select "Edit" a menu 15 will appear. A user may highlight entries in menu 15 by depressing button 27 on mouse 20. Upon releasing button 27 information about the highlighted entry will appear. For instance if a user selects "Delete" a window 7 containing information about "Delete" will appear as shown in Figure 8. In addition, a command bar 21 allows a user other options. For instance, the user may print the information in window 9 by selecting "Print". The user may be selecting "Index" recall selection box 30 shown in Figure 5. The user may remove window 9 from being displayed by selecting "Hide". The user may view a list of related information by selecting "Related Topics."

In addition to the methods listed above, applications 11, 12 and 13 may call main help module 16 under other special circumstances. For instances error messages may include a help box. An example, is shown in Figure 9. An error message 8 appeared in window 1 with a box entitled "Help". Using cursor 2 to select the "Help" box results in the display of a window 9 which more fully explains the error message. In addition, a command bar 22 allows a user other options.

Main help module may be called by a number of application programs. Figures 10 through 16 show data flow through main help module 16 when called by a single application, application program 11.

Each application has associated with it a help file. The format of help files are the same for every application. Each help file, however, has text particular to its associated application program. In this way, Main Help Module 16 can utilize help files for different applications.

In Figure 10, main help module 16 receives messages from application program 11 through a data channel 94 and from user input/output devices 81 through a data channel 95. Main help module 16 also access information in help file 82 through a data channel 84. A separate help file exists for each application. Help file 82 contains text which provides help information for a user of application program 11. Main help module 16 sends help display data to

system intrinsics 17 through a data channel 97, and sends window display data to system intrinsics 17 through a data channel 98.

In Figure 11, main help module 16 is shown to include a message front end routine 91, a command processor routine 92 and a display method routine 93. Message front end routine 91 sends application messages to command processor routine 92 through data channel 99. Command processor routine 92 sends current application to display method routine 93 through a data channel 96 and sends an index offset to display method routine 93 through a data channel 100.

In Figure 12, display method routine 93 is shown to include a load help file routine 121 and an output text routine 122. Load help file routine 121 buffers output which it sends to output text routine 122 through data channel 123.

In Figure 13, output text routine 122 is shown to include a process character routine 130, a create font routine 127 and an execute end method routine 134. Process character routine 130 sends normal characters to a text output buffer file 132 through a data channel 137 and sends index offset information to a rectangle table file 131 through a data channel 138. Process character routine 130 also sends font information to create font routine 127 through a data channel 129 and sends information indicating end method type to execute end method routine 134 through a data channel 135.

Create font routine 127 sends fonts to execute end method routine 134 through a data channel 128. Execute end method routine 134 receives text from text output buffer file 132 through a data channel 139 and sends information containing a rectangle size to rectangle table file 131 through a data channel 136.

In Figure 14, command processor routine 92 is shown to include a message dispatch 149, a scroll routine 141, a context switch routine 142, a register routine 143, an un-register routine 144, a topic routine 145, an index routine 146, a related topic routine 147 and a user command routine 148. Message dispatch 140 sends scroll commands to scroll routine 141 through a data channel 150. Message dispatch 140 sends change connection commands to context switch routine 142 through a data channel 151. Message dispatch 140 sends register commands to register routine 143 through a data channel 152. Message dispatch 140 sends un-register commands to un-register routine 144 through a data channel 153. Message dispatch 140 sends topic commands to topic routine 145 through a data channel 154. Message dispatch 140 sends index commands to index routine 146 through a data channel 155. Message dispatch 140 sends related topic commands to related topic routine 147 through a data channel 156. Message dispatch 140 sends user message commands to user command routine 148 through a data channel 157.

An application table 158 receives output from register routine 143 and un-register routine 144 and us accessed by context switch routine 142. Topic routine 145 and index routine 1446 both access help file 82. Related topic routine accesses rectangle table 131.

In Figure 15 user command routine 148 is shown to include a dispatch user command routine 161, a change window size routine 162, an index routine 163 and a pop topic routine 164. Dispatch user command routine sends change window size commands to change window size routine through a data channel 165. Dispatch user command routine 161 sends index commands to index routine 163 through a data channel 166. Dispatch user command routine 161 sends a command to retrieve a previous command to pop topic routine 164 through a data channel 167. Index routine 163 retrieves an index table from help file 82 through a data channel 168. Pop topic routine 164 accesses previous commands from topic stack 160 through a data channel 169.

In Figure 16, change window size routine 162 is shown to include a compute size offset routine 170 and a move windows routine 171. Compute size offset routine 170 sends a size offset to move windows routine 171 through a data channel 172.

In Figure 17 the structure of help file 82 is shown. Help file 82 contains a file header 173, a section 174 containing a plurality of text headers accompanied by text, a section 175 containing a context table, and a section 176 containing an index table. File header 173 includes a pointer 179 to the first text header in section 174, a pointer 178 to the first context table entry in section 175 and a pointer 177 to the first index table entry in section 176.

In Figure 18 additional structure of help file 82 is shown. Section 174 contains a plurality of text headers, each text header being immediately followed by text. For instance, a text header and text 181, a text header and text 182, a text header and text 183, a text header and text 184, a text header and text 185 and a text header and text 186 are shown. Each text header and text are used for display to a user. A text header includes a topic name, a length of the following text and pointers to text headers for related topics. For instance, in window 9 shown in Figure 9, there is displayed the topic name "Disc Full" and the text "This error..." which follows.

The context table in section 175 is used by main help module 16 to locate text on a topic requested by user. A context entry 188 includes a pointer to a text header and text 185, and a context entry 189 includes a pointer to a text header and text 182.

The index table in section 176 is used by main help module 16 to index help text headers and text entries when a user users selection box 20 shown in Figure 5. Each entry in the index table includes a pointer to a text header. For instance, an index entry 190 includes a pointer to text header and text 181. An index entry 191 includes a pointer to text header and text 184.

A computer program which is a complete embodiment of main help module 16 is included as appendix A.

Claims

1. A computing system in which a user may interact with a plurality of applications

(11,12,13) which run thereon, said system having a help system comprising:
a plurality of help text files (82) wherein each help text file in the plurality of help text files is associated with an application in the plurality of applications (11,12,13) and,
a main help routine (16), which may be accessed by a user from every application in the plurality of applications (11,12,13) and which presents a user with help information for a first application in the plurality of applications by accessing data from a first help text file in the plurality of first help text files, wherein the first help text file is associated with the first application.

2. A system according to claim 1 wherein each help text file (82) comprises:
a plurality of entries (174) each entry including a header, the header including a topic name, and each entry including text containing information about the topic named in the topic name.

3. A computing system having a system memory (10), a display (14) and a user help system comprising:
a plurality of application programs (11,12,13) residing in the system memory and which may be run by a user;
a plurality of help text files (82) residing in the system memory (10) each help text file from the plurality of text files being associated with an application program from the plurality of application programs (11,12,13); and
a main help routine (16) residing in the system memory (10) which may be accessed by the user from every application in the plurality of applications (11,12,13) and which presents the user with help information for a first application in the plurality of applications (11,12,13) by accessing data from a first help text file in the plurality of first help text files (87) and displaying the information on the display (14) wherein the first help text file is associated with the first application.

4. A system according to claim 3 wherein each help text file comprises:
a plurality of entries (174) each entry including a header, the header including a topic name, and each entry including text containing information about the topic named in the topic name.

5. A method for providing help information to a user from a plurality of applications (11,12,13), the method comprising:

(a) developing for each application in the plurality of applications an associated help text file (82) which contains help information for the application;

(b) using a single main help routine (16) to present a single user interface which is accessible for each application in the plurality of applications; and

(c) accessing the help text file (82) associated with a current running application for help information when a user requests help information from the current running application.

6. A method according to claim 5 wherein step (c) includes accessing and displaying a topic name and text.

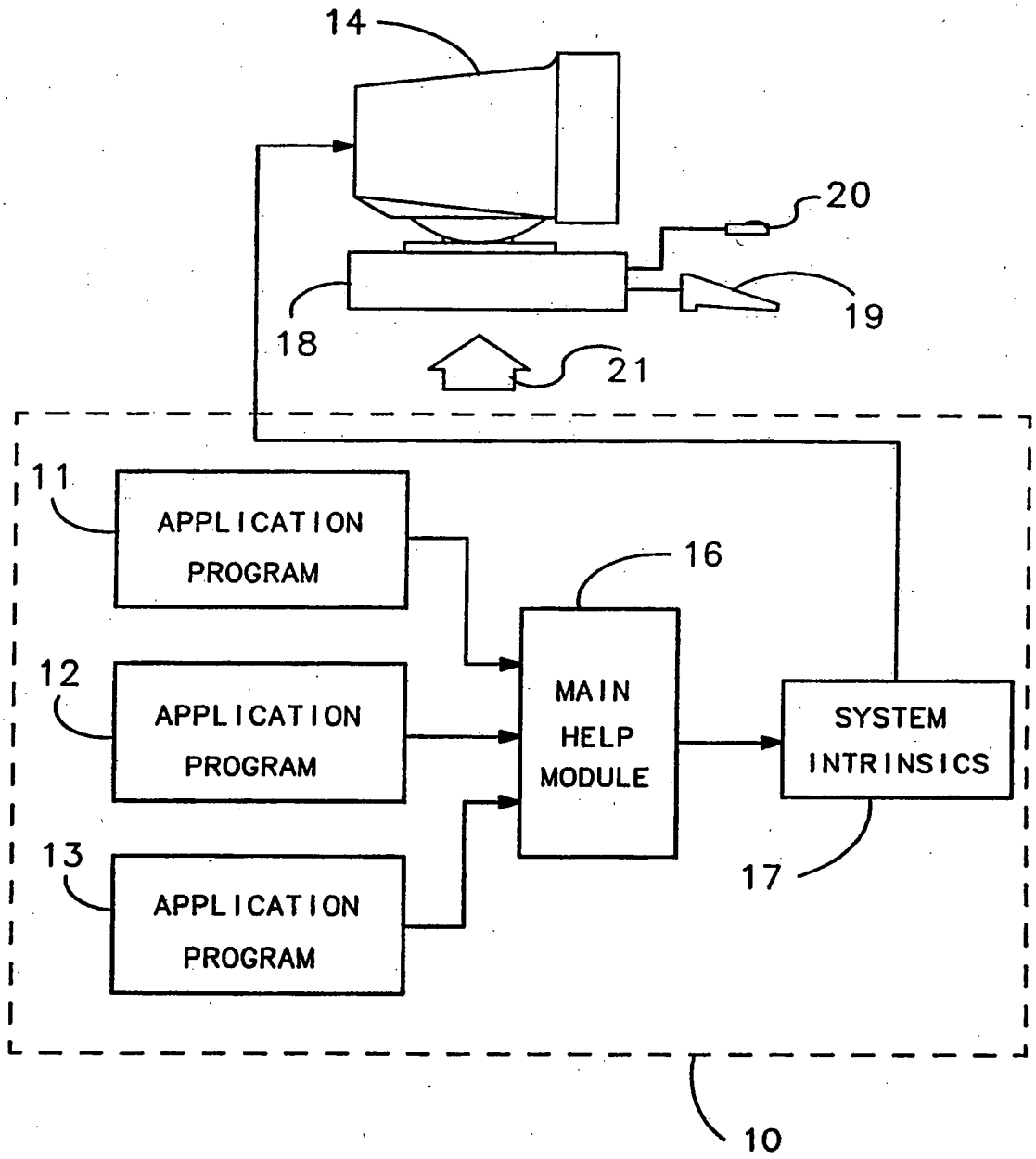
14
20
19
18
21
11
APPLICATION PROGRAM
12
APPLICATION PROGRAM
13
APPLICATION PROGRAM
16
MAIN HELP MODULE
SYSTEM INTRINSICS
17
10

FIG 1

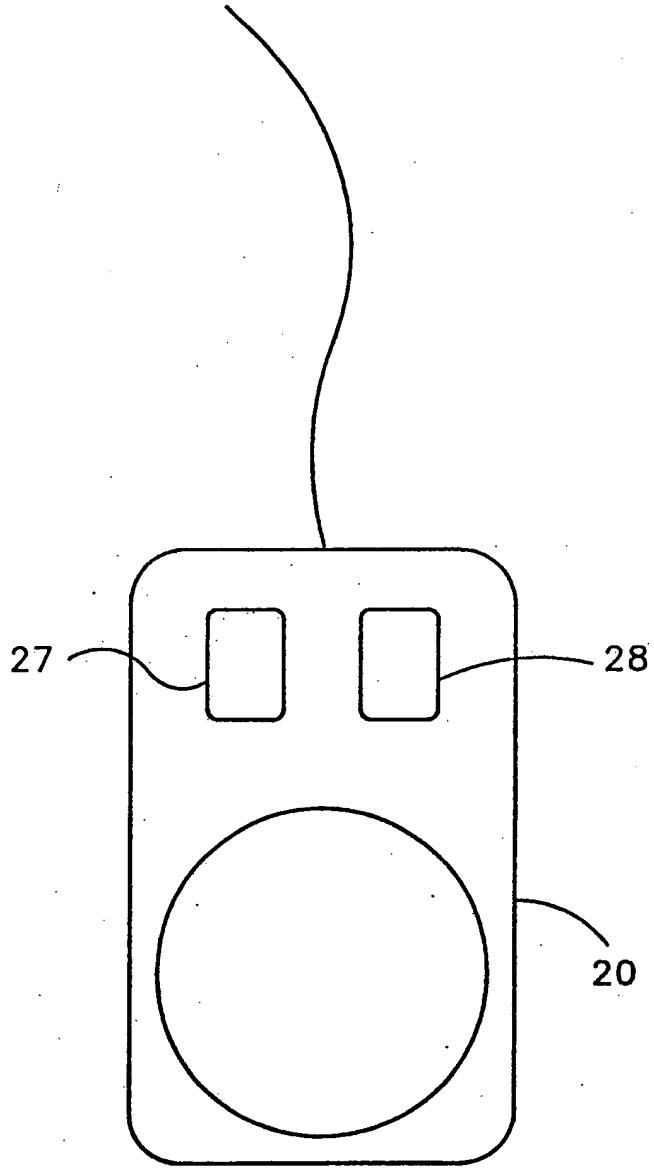

FIG 2

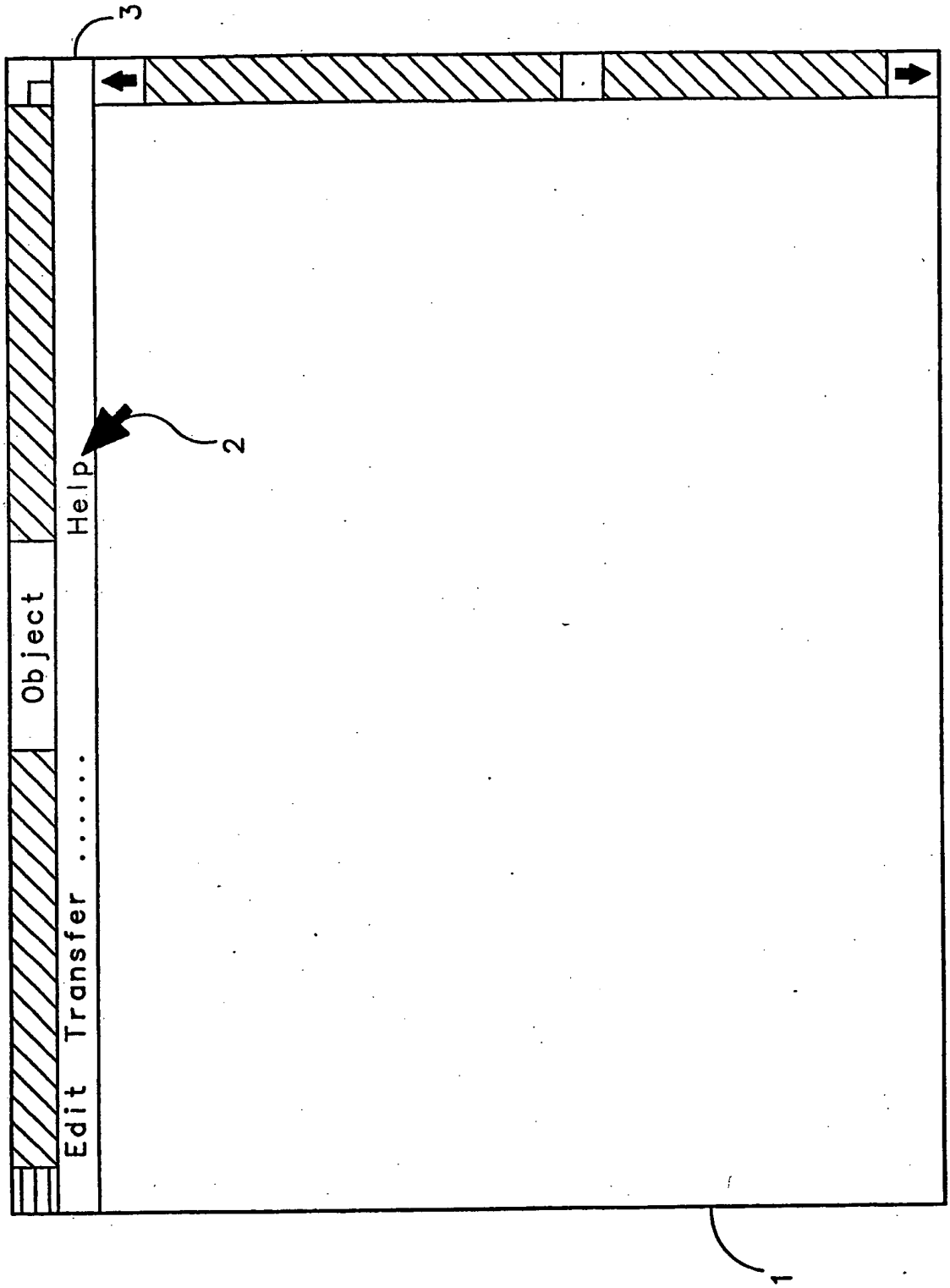

FIG 3

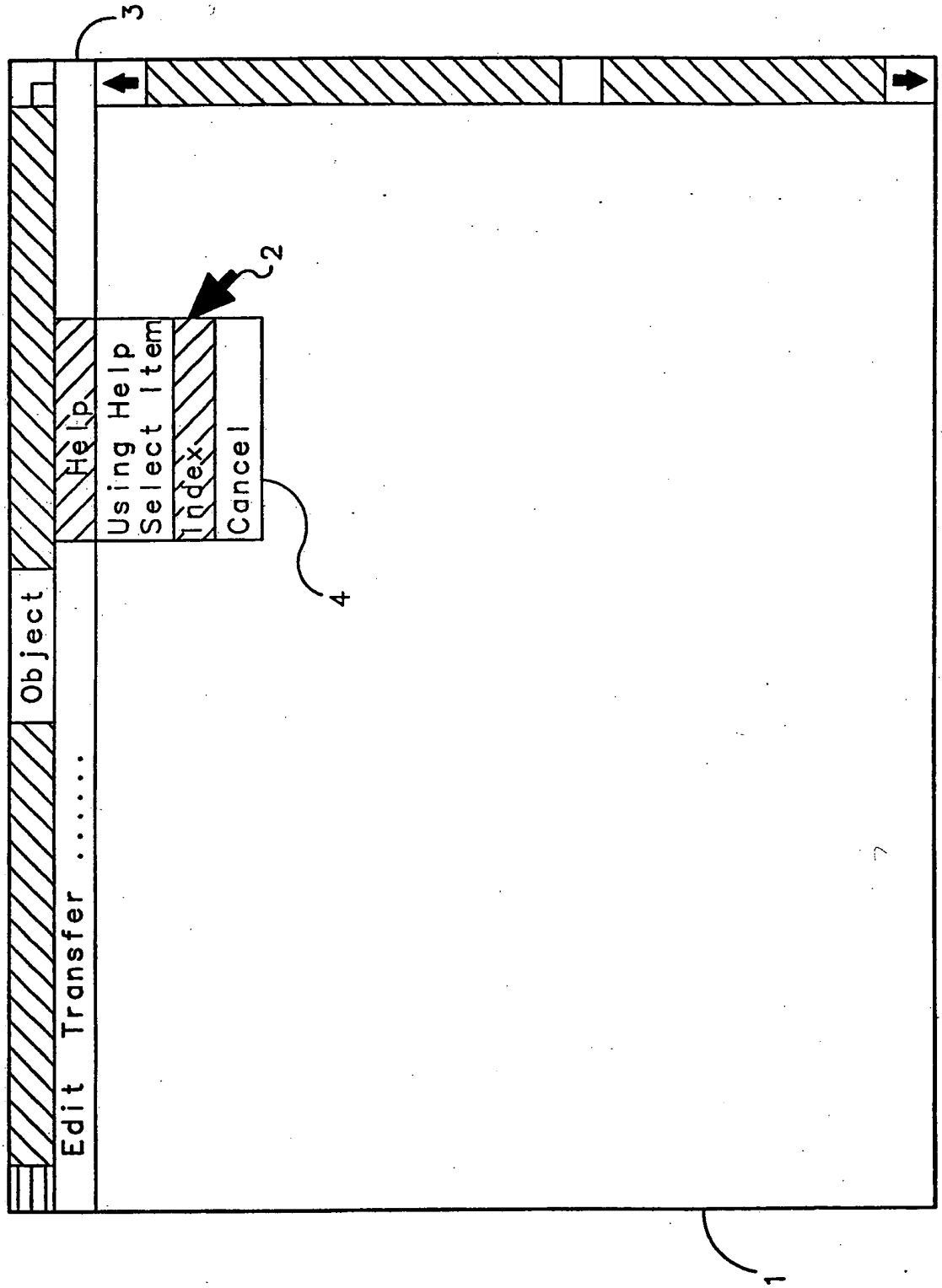
3
2
Help
Using Help
Select Item
Index
Cancel
4
Object
Edit Transfer .....
1

FIG 4

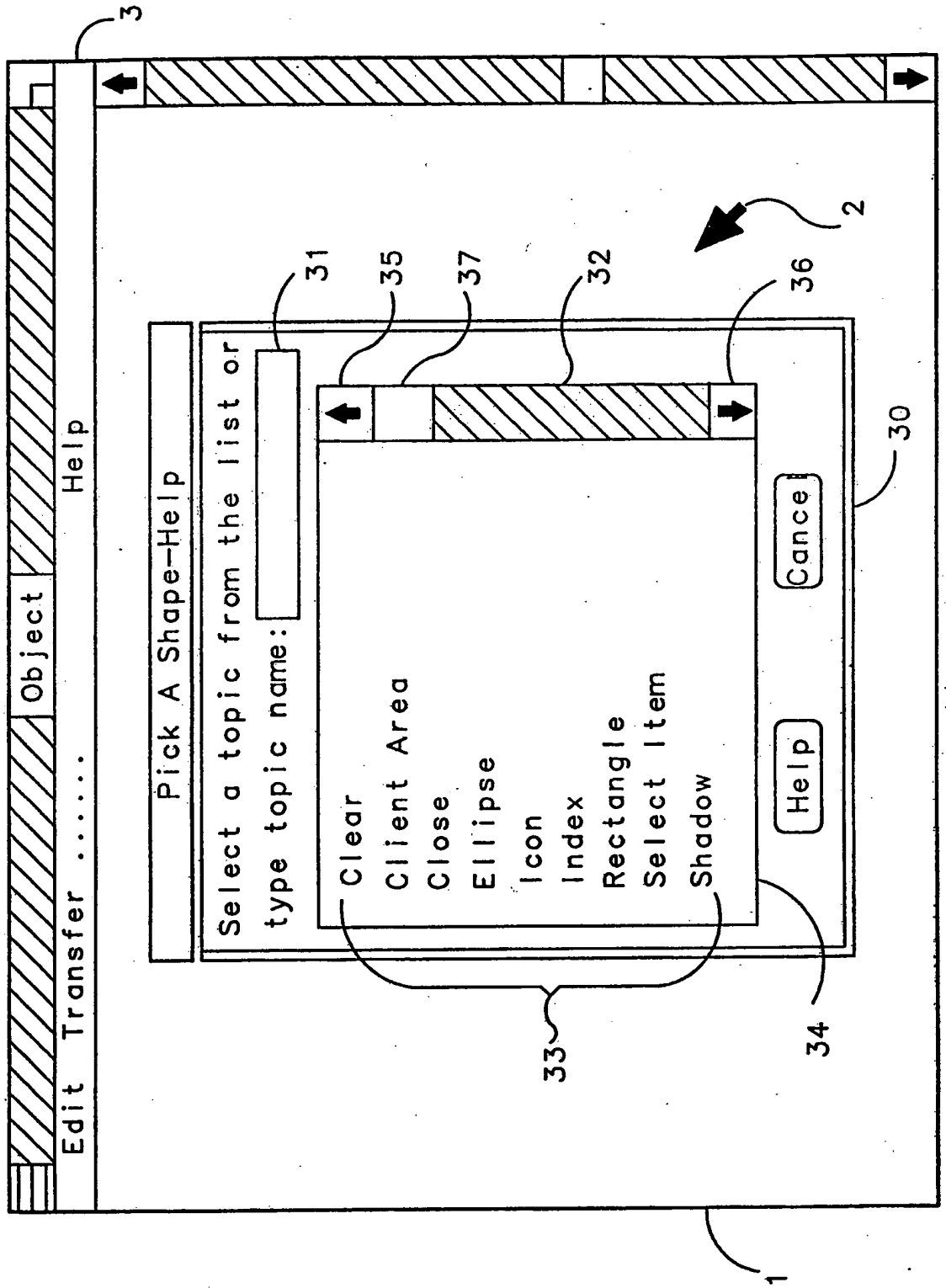
3
Edit Transfer .....
Object
Help
Pick A Shape-Help
Select a topic from the list or
type topic name:
31
35
37
32
36
2
30
Clear
Client Area
Close
Ellipse
Icon
Index
Rectangle
Select Item
Shadow
33
34
Cancel
Help
1

FIG 5

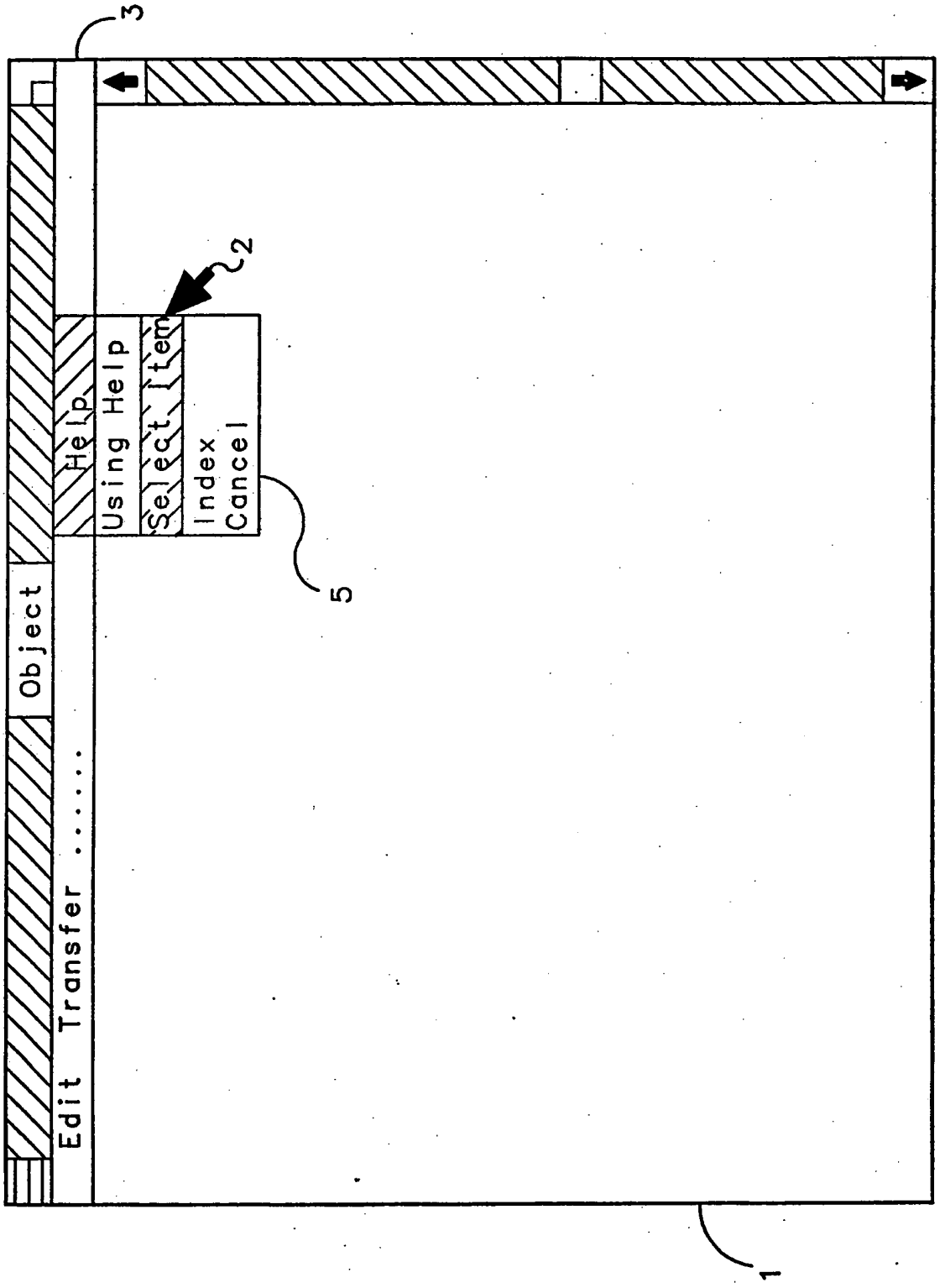
3
2
Edit Transfer .....
Object
Help
Using Help
Select Item
Index
Cancel
5
1

FIG 6

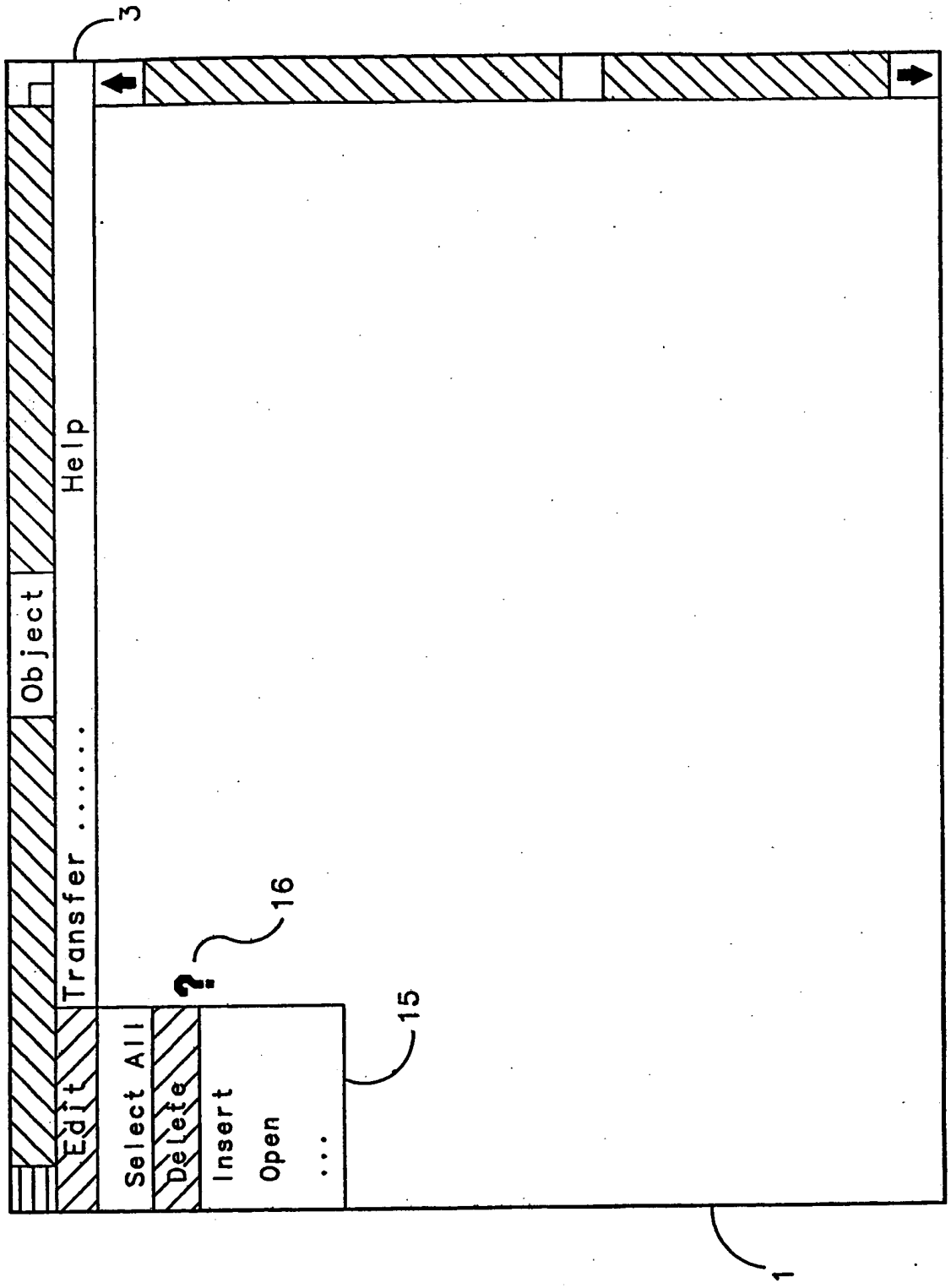
3
Help
Object
Transfer .....
Edit
Select All
Delete
Insert
Open
...
?
16
15
1

FIG 7

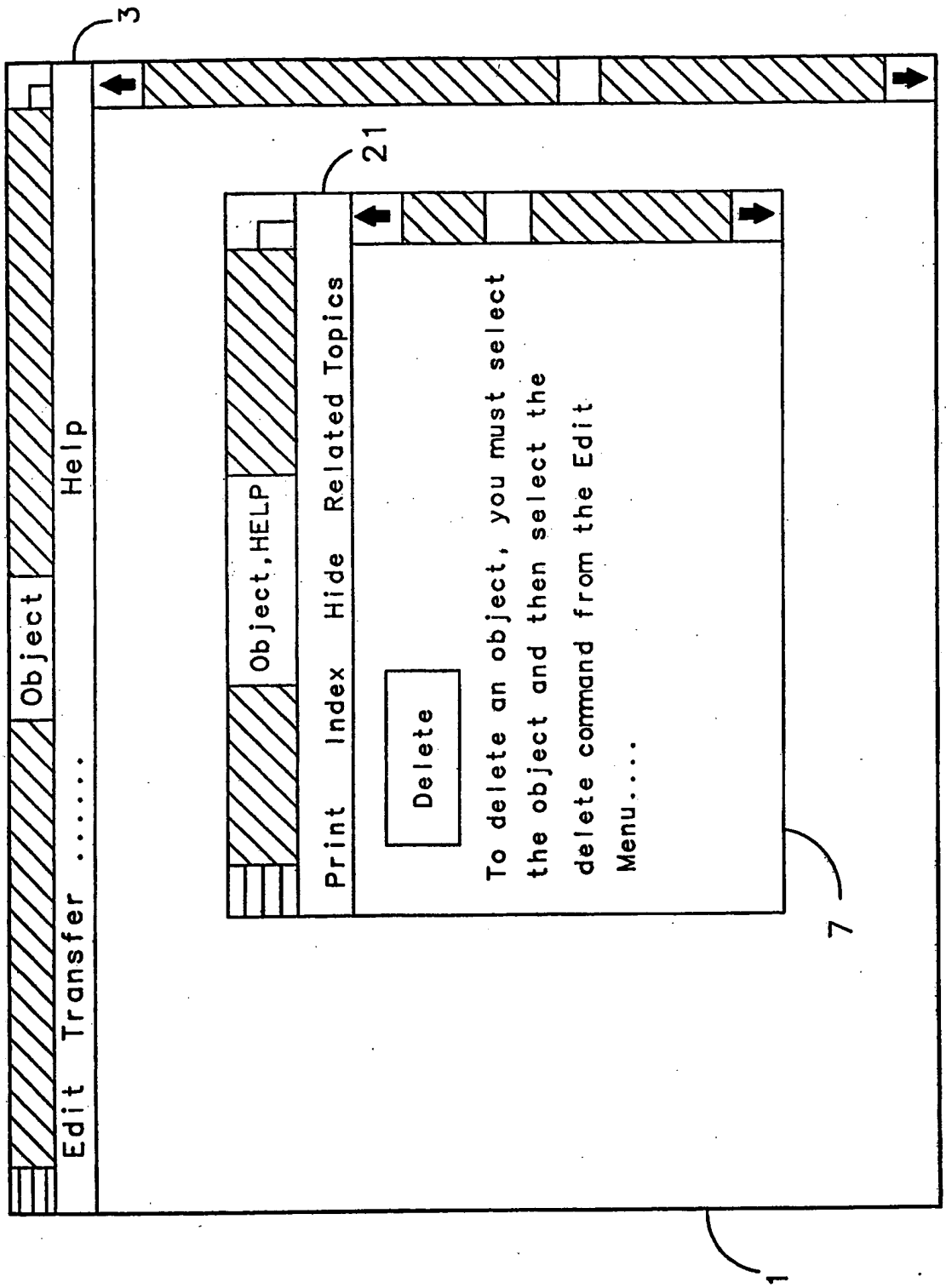
3
Edit Transfer .....
Object
Help
21
Object,HELP
Print Index Hide Related Topics
Delete
To delete an object, you must select
the object and then select the
delete command from the Edit
Menu....
7
1

FIG 8

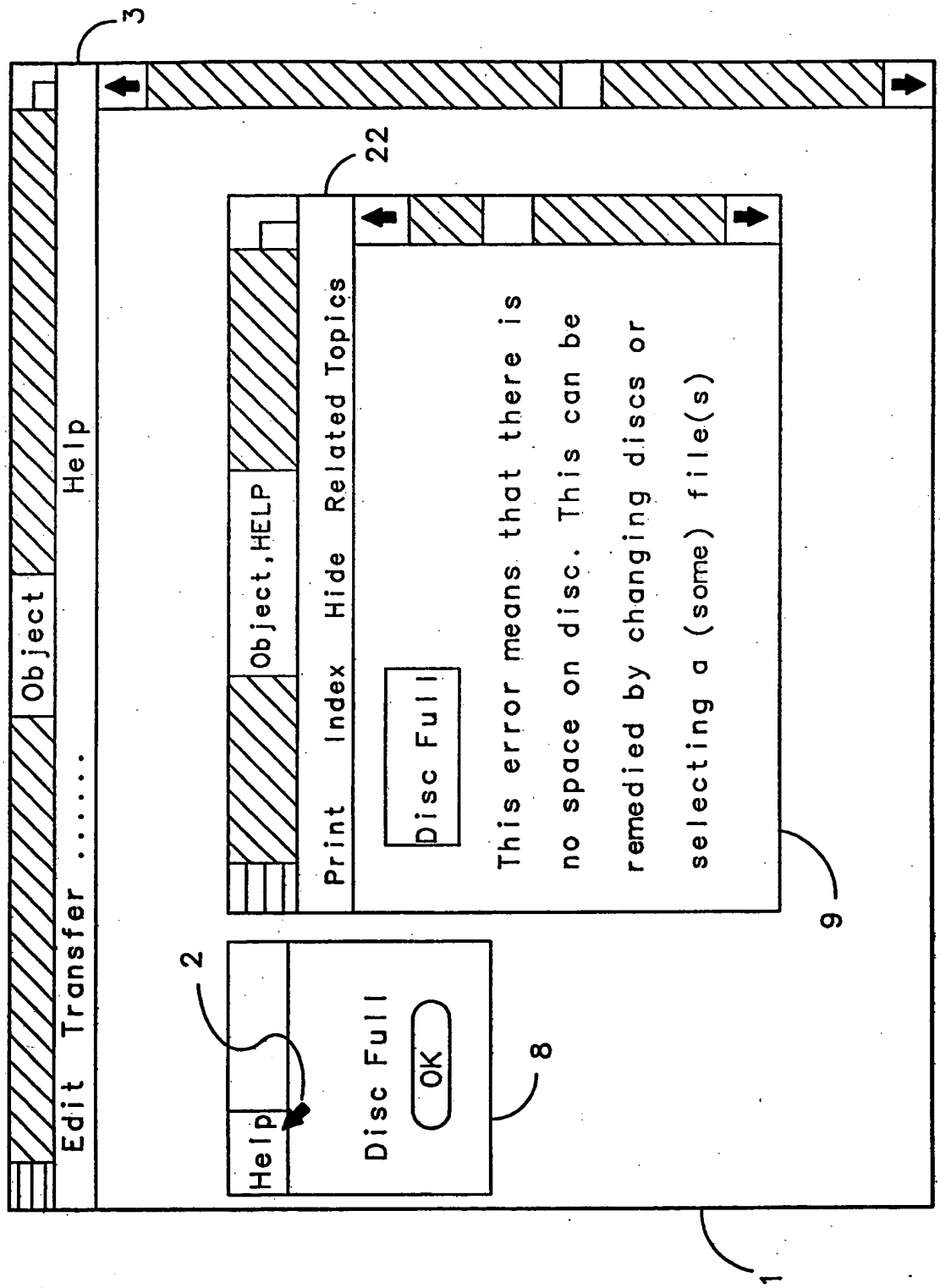
3
Edit Transfer ..... Help
Object
22
Print Index Hide Related Topics
Object,HELP
Disc Full
This error means that there is no space on disc. This can be remedied by changing discs or selecting a (some) file(s)
9
2
Help
Disc Full
OK
8
1

FIG 9

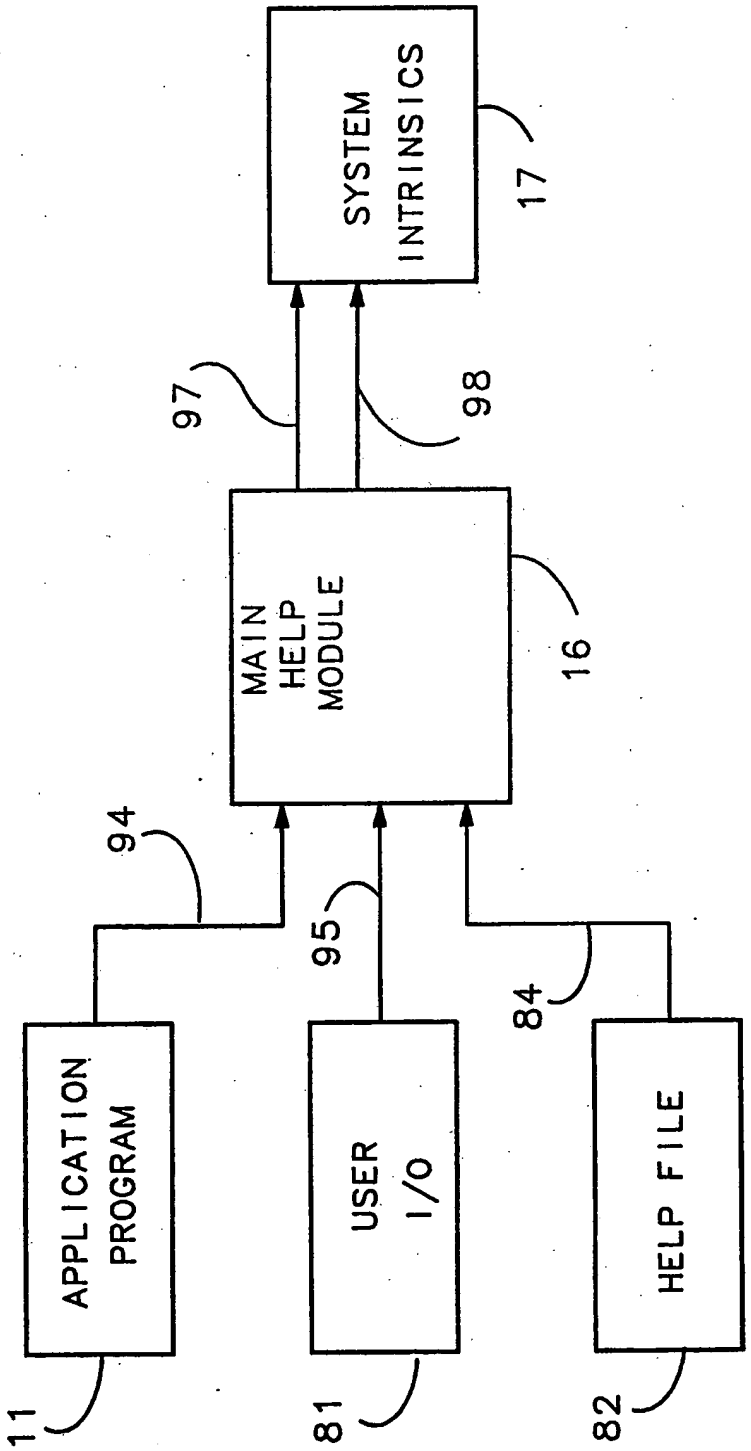
SYSTEM INTRINSICS
17
97
98
MAIN HELP MODULE
16
94
95
84
APPLICATION PROGRAM
USER I/O
HELP FILE
11
81
82

FIG 10

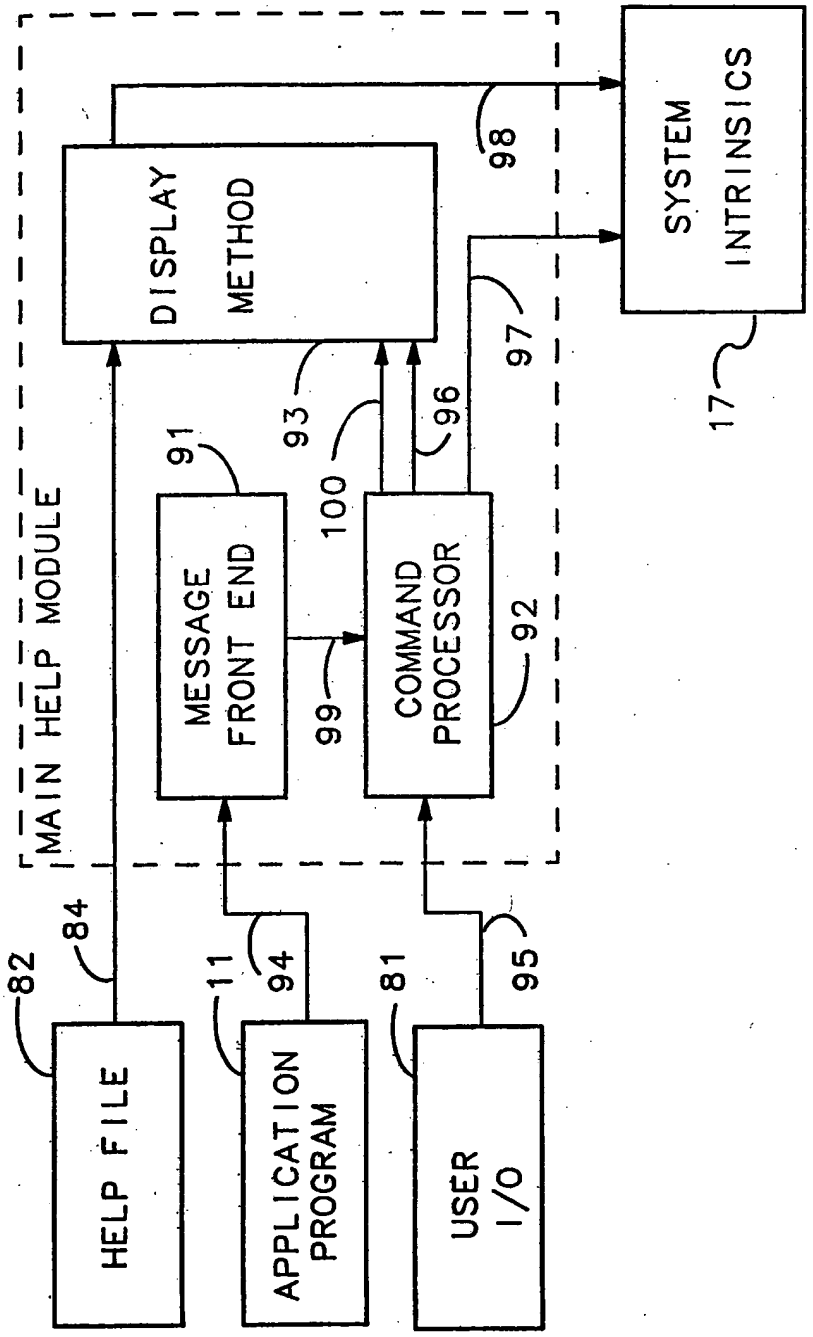
MAIN HELP MODULE
DISPLAY METHOD
MESSAGE FRONT END
COMMAND PROCESSOR
SYSTEM INTRINSICS
HELP FILE
APPLICATION PROGRAM
USER I/O
82
84
11
94
81
95
91
93
100
96
97
98
99
92
17

FIG 11

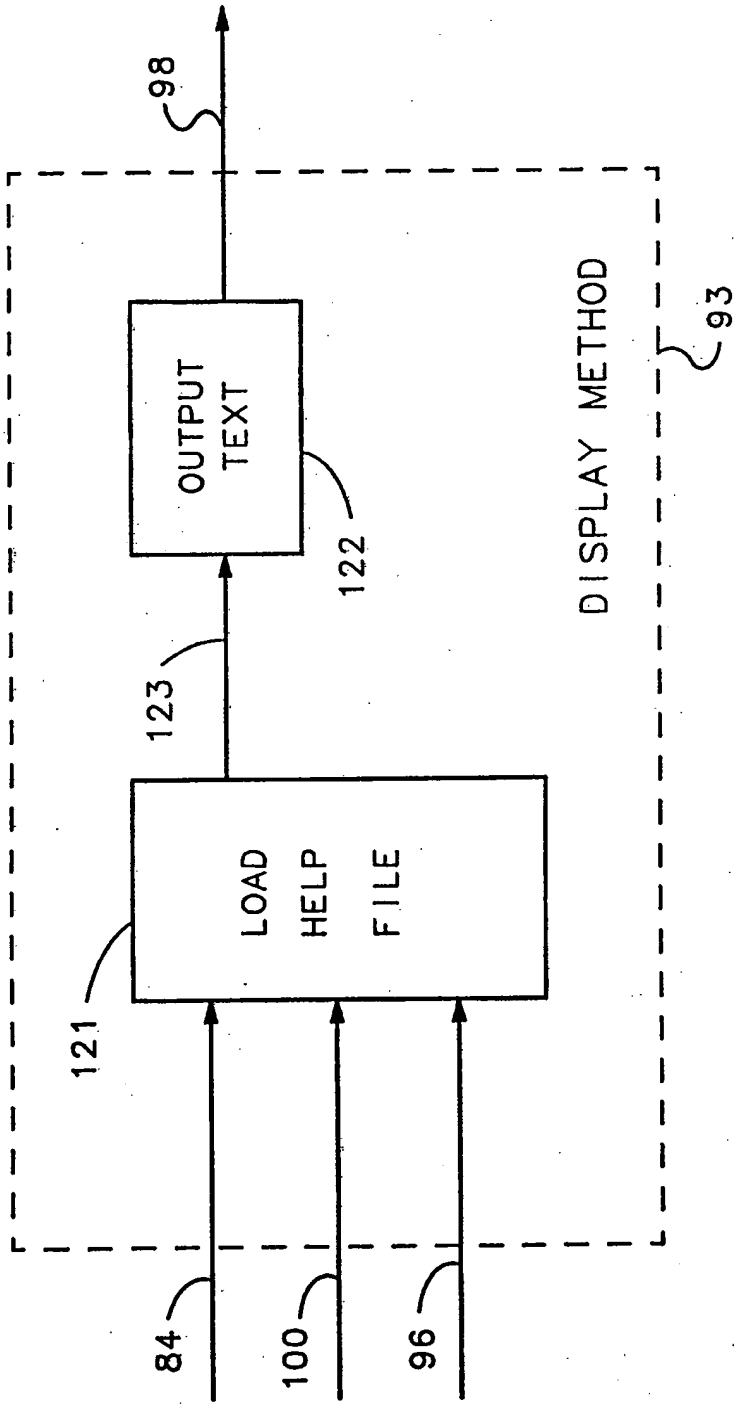
98
OUTPUT TEXT
122
123
LOAD HELP FILE
121
DISPLAY METHOD
93
84
100
96

FIG 12

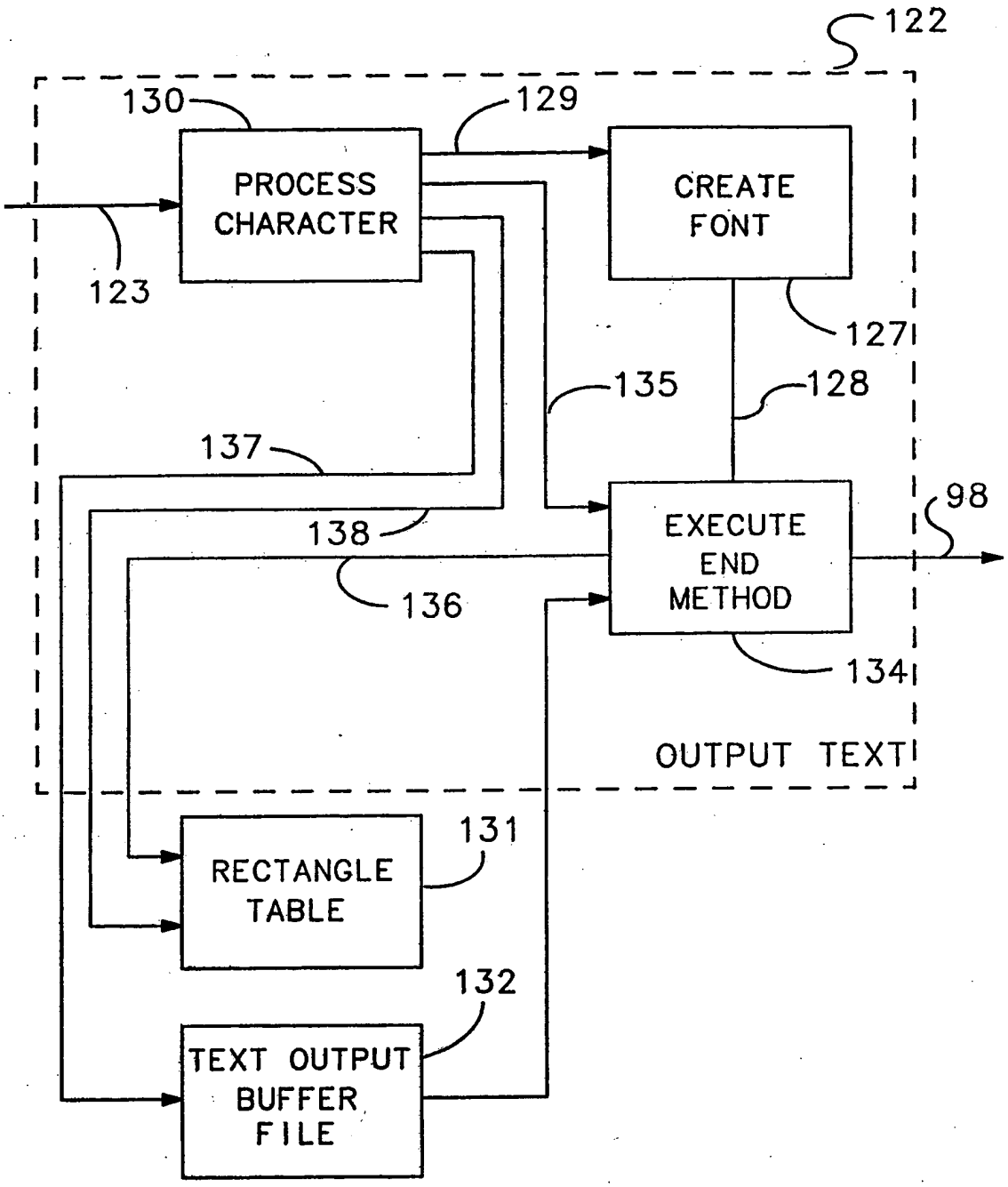
122
130
129
PROCESS CHARACTER
CREATE FONT
123
127
135
128
137
98
138
EXECUTE END METHOD
136
134
OUTPUT TEXT
131
RECTANGLE TABLE
132
TEXT OUTPUT BUFFER FILE

FIG 13

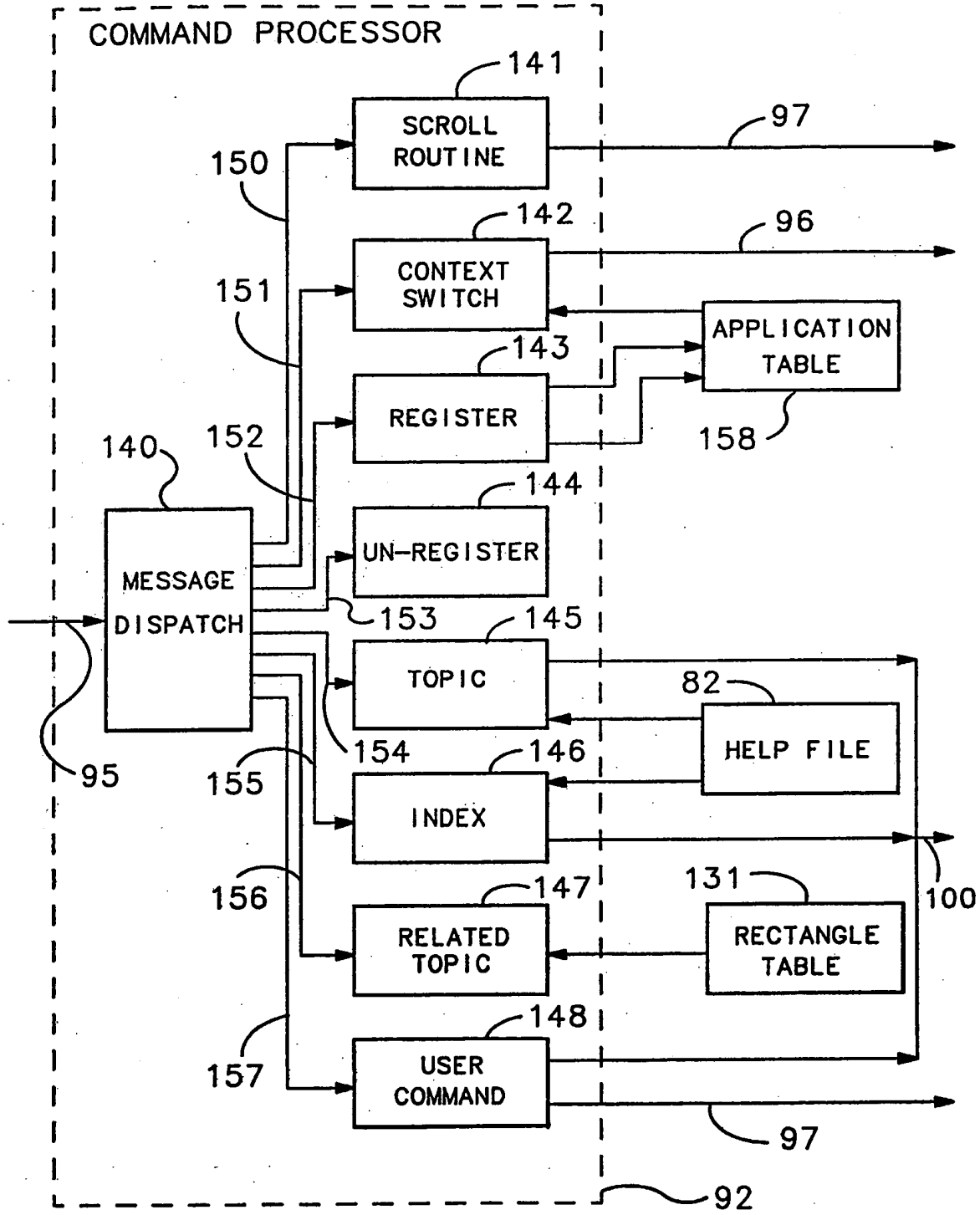
COMMAND PROCESSOR
141
SCROLL ROUTINE
97
150
142
CONTEXT SWITCH
96
151
APPLICATION TABLE
143
REGISTER
158
152
140
144
MESSAGE DISPATCH
UN-REGISTER
153
145
TOPIC
82
HELP FILE
95
155
154
146
INDEX
156
131
147
RELATED TOPIC
RECTANGLE TABLE
100
148
157
USER COMMAND
97
92

FIG 14

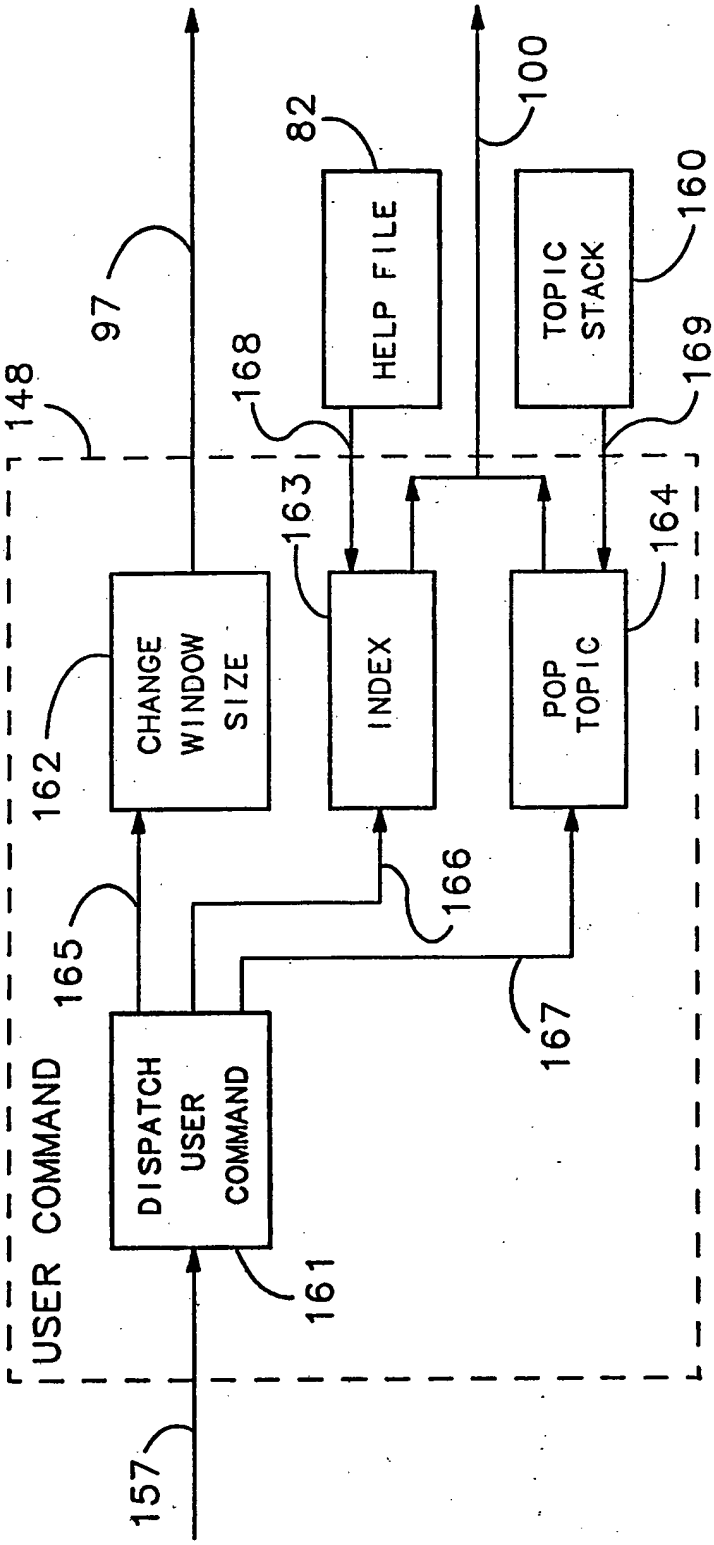
USER COMMAND
DISPATCH USER COMMAND
CHANGE WINDOW SIZE
INDEX
POP TOPIC
HELP FILE
TOPIC STACK
148
157
161
162
163
164
165
166
167
168
169
160
82
100
97

FIG 15

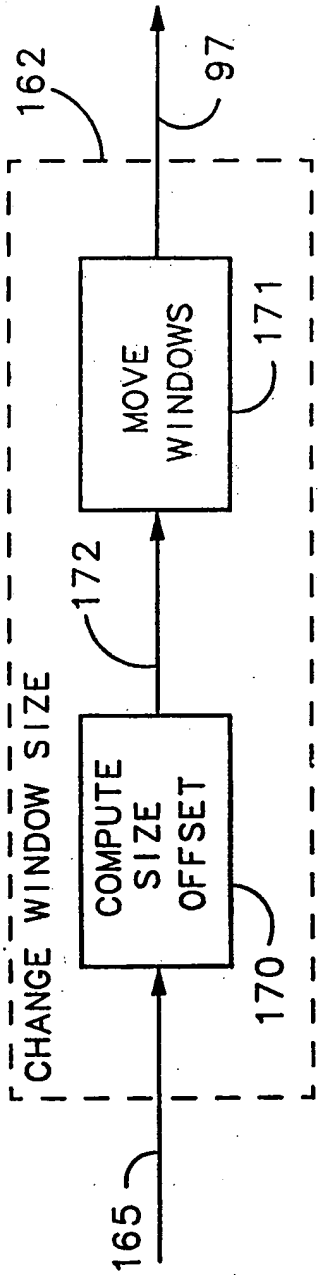
CHANGE WINDOW SIZE
COMPUTE SIZE OFFSET
MOVE WINDOWS
162
165
170
171
172
97

FIG 16

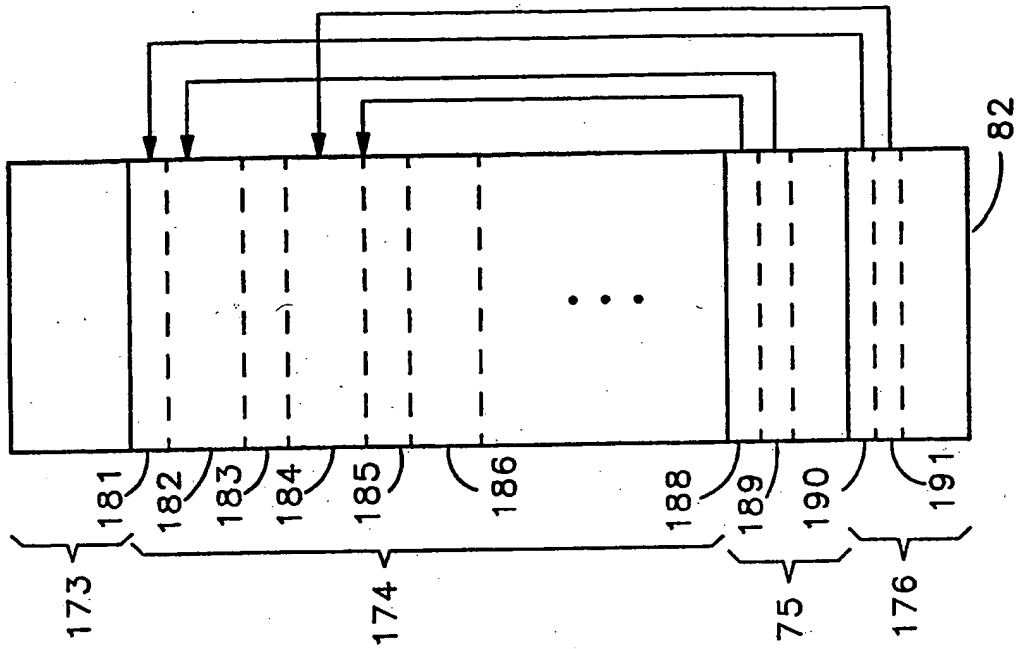
82
181
182
183
184
185
186
188
189
190
191
173
174
175
176

FIG 18

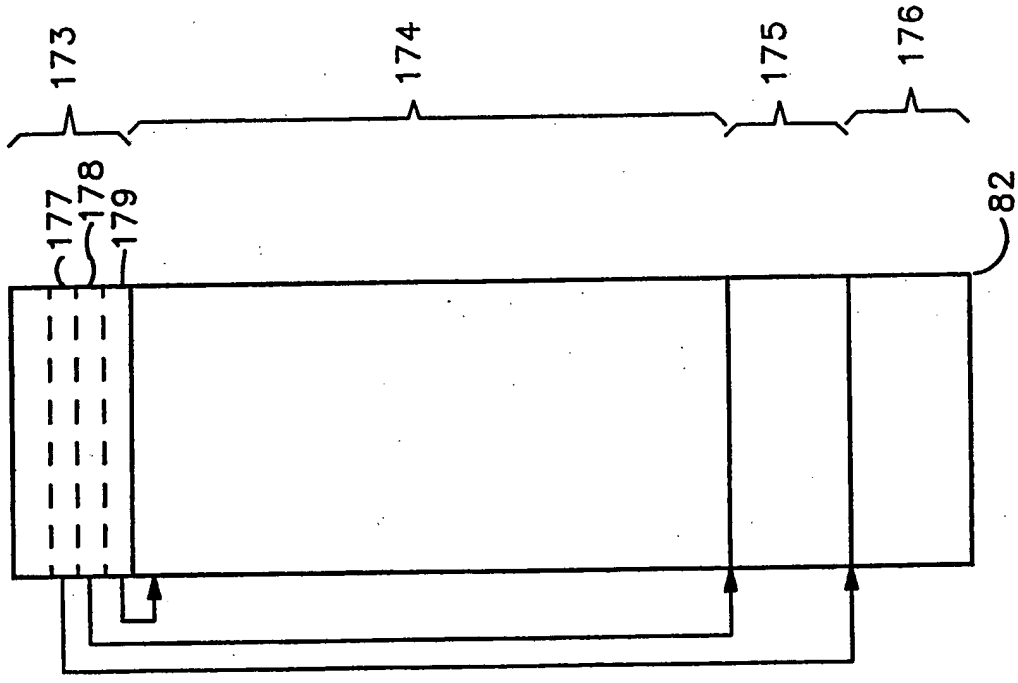
173
174
175
176
177
178
179
82

FIG 17